REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 31-08-2012	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 1-Jun-2009 - 31-May-2012

4. TITLE AND SUBTITLE Partial Planning Reinforcement Learning: Final Report	5a. CONTRACT NUMBER W911NF-09-1-0153
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 611102
6. AUTHORS Prasad Tadepalli, Alan Fern	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Oregon State University Office of Sponsored Programs Oregon State University Corvallis, OR 97331 -2140	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 55667-NS.11

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

This project explored several problems in the areas of reinforcement learning, probabilistic planning, and transfer learning. In particular, it studied Bayesian Optimization for model-based and model-free reinforcement learning, transfer in the context of model-free reinforcement learning based on hierarchical Bayesian framework, probabilistic planning based on monte-carlo tree search, and new algorithms for learning task hierarchies. The algorithms were empirically evaluated in real-time strategy games and other standard benchmark tasks and were

15. SUBJECT TERMS

Reinforcement Learning, Bayesian Optimization, Active Learning, Action Model Learning, Decision Theoretic Assistance

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Prasad Tadepalli
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 541-737-5552

Report Title

Partial Planning Reinforcement Learning: Final Report

ABSTRACT

This project explored several problems in the areas of reinforcement learning, probabilistic planning, and transfer learning. In particular, it studied Bayesian Optimization for model-based and model-free reinforcement learning, transfer in the context of model-free reinforcement learning based on hierarchical Bayesian framework, probabilistic planning based on monte-carlo tree search, and new algorithms for learning task hierarchies. The algorithms were empirically evaluated in real-time strategy games and other standard benchmark tasks and were shown to perform better than the state of the art approaches. The project also developed new theoretical frameworks for learning deterministic action models and for decision theoretic assistance and proved new formal results in these areas. The project helped graduate two Ph.D. students and partially funded the research of two other students.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received	Paper
2011/08/29 19 4	Neville Mehta, Soumya Ray, Prasad Tadepalli, Thomas G. Dietterich. Automatic Discovery and Transfer of Task Hierarchies in Reinforcement Learning, Association for the Advancement of Artificial Intelligence, (04 2011): 35. doi:

TOTAL: 1

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations

1. Prasad Tadepalli, Planning and Reinforcement Learning: A Tale of Two Worlds, presented at Bellairs workshop on Model-based Reinforcement Learning, Spring 2009
2. Neville Mehta, Learning and Planning with Partial Models, presented at Bellairs workshop on Model-based Reinforcement Learning, Spring 2009

Number of Presentations: 2.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
2012/08/31 1(	10	Ronald Bjarnason, Alan Fern, Prasad Tadepalli. Lower Bounding Klondike Solitaire with Monte-Carlo Planning, International Conference on Automated Planning and Scheduling. 2009/09/19 03:00:00, . : ,
2012/08/31 1(	8	Alan Fern, Prasad Tadepalli, Aaron Wilson. Bayesian Policy Search for Multi-agent Role Discovery, National Conference on Artificial Intelligence. 2010/07/11 03:00:00, . : ,
2012/08/22 1(	9	Alan Fern, Aaron Wilson, Prasad Tadepalli. Incorporating Domain Models into Bayesian Optimization for RL, European Conference on Machine Learning. 2010/09/20 03:00:00, . : ,
2012/08/22 1!	7	Alan Fern, Prasad Tadepalli. A Computational Decision Theory for Interactive Assistants, AAAI Workshop on Interactive Game Theory and Decision Theory. 2010/07/10 03:00:00, . : ,
2012/08/22 1!	6	Alan Fern, Prasad Tadepalli. A Computational Decision Theory for Interactive Assistants, Neural Information Processing Systems. 2010/12/06 03:00:00, . : ,
2012/08/22 1‹	5	Prasad Tadepalli, Alan Fern, Neville Mehta. Autonomous Learning of Action Models for Planning, Neural Information Processing Systems. 2011/12/12 03:00:00, . : ,
2011/08/29 1!	3	Neville Mehta, Prasad Tadepalli, Alan Fern. Efficient Learning of Action Models for Planning, Workshop on Planning and Learning at ICAPS 2011. 2011/06/13 03:00:00, . : ,
2011/08/29 1!	1	Aaron Wilson, Alan Fern, Prasad Tadepalli. Transfer Learning in Sequential Decision Problems: A Hierarchical Bayesian Approach, Workshop on Unsupervised and Transfer Learning at ICML 2011. 2011/07/02 03:00:00, . : ,
2011/08/29 1!	2	Aaron Wilson, Alan Fern, Prasad Tadepalli. A Behavior-based Kernel for Policy Search via Bayesian Optimization, Workshop on Planning and Acting with Uncertain Models, ICML 2011. 2011/07/02 03:00:00, . : ,

TOTAL: **9**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received Paper

TOTAL:

Number of Manuscripts:

Books

Received Paper

TOTAL:

Patents Submitted

Patents Awarded

Awards

The following paper received best student paper award at the International Conference on Automated Planning and Scheduling, 2009.

Bjarnason, R., Fern, A. and Tadepalli, P., Lower Bounding Klondike Solitaire with Monte Carlo Planning, ICAPS, 2009.

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Aaron Wilson	1.00	
Nevile Mehta	0.58	
Kshitij Judah	0.28	
Robin Hess	0.08	
	0.00	
FTE Equivalent:	**1.94**	
Total Number:	**4**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Prasad Tadepalli	0.05	
Alan Fern	0.01	
FTE Equivalent:	**0.06**	
Total Number:	**2**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 0.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 0.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 0.00

Names of Personnel receiving masters degrees

NAME	
Total Number:	

Names of personnel receiving PHDs

NAME	
Aaron Wilson	
Neville Mehta	
Total Number:	**2**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

1. We developed and evaluated UCT-based Monte Carlo planning algorithms in Solitaire and tactical battles in real-time strategy games. Our approach to Solitaire yielded the first known lower bound of 35% success rate.
2. We implemented algorithms for automatic discovery of roles for agents using Bayesian policy search and successfully evaluated in tactical battles in RTS games. The role-specific policies were shown to transfer to similar domains.
3. We developed new algorithms for learning task hierarchies for reinforcement learning from demonstrations. The task hierarchies are shown to improve convergence speed of reinforcement learning in a number of domains.
4. We designed a model-based reinforcement learning algorithm based on Bayesian optimization and showed significantly faster convergence in a number of benchmark reinforcement learning domains.
5. We developed a model-free reinforcement learning algorithm based on Bayesian Optimization (BO) that takes into account trajectory information and significantly improves upon previous model-free BO methods for RL.
6. We formalized the notion of partial action models and developed theory and algorithms for learning such models and planning with them.
7. We showed new hardness results and proved performance bounds for the effectiveness of myopic heuristics on the problem of decision theoretic assistance.

Technology Transfer

Partial Planning Reinforcement Learning: Final Report
Proposal Number 55667-NS

Prasad Tadepalli and Alan Fern, Oregon State University

1. Executive Summary

The objective of our research was to make novel advances in decision-theoretic planning, which is studied by two different communities, namely, automated planning and reinforcement learning. Both these paradigms have certain limitations. In particular, the work in AI planning generally assumed that the domain dynamics can be compactly described using a declarative action description language. It also assumed that such action descriptions are readily available to the planner. Both of these assumptions break down in complex stochastic domains. On the other hand, most work in reinforcement learning was focused on learning from simulators by trial and error, which is inherently inefficient. Model-free reinforcement learning ignored the aspect of reasoning about action models to perform better. Our objective was to develop new algorithms that explored the wide spectrum between the two extremes to reap the benefits of both these fields while overcoming their respective limitations. Our contributions fall into the following 4 areas. The results are expanded further in the main body of the report.

- **Bayesian Reinforcement Learning**: We developed new more powerful approaches to reinforcement learning (RL) by formulating the RL problem in the hierarchical Bayesian framework. The principal advantage of doing this is that it allows the learned knowledge to transfer from one task to a related task. We developed directed exploration algorithms based on Bayesian Optimization that take explicit account of the trajectory information. We also developed broadly useable computational infrastructure to study RTS games.
- **Learning Task Hierarchies and Action Models**: We developed new methods for designing task hierarchies by *learning* them from example traces of the task execution. This approach required us to know the action models to analyze the causal relations between the different actions in the trace. Hence we explored the problem of learning actions models and planning with them. During learning, the action models are partial or incomplete in that they only describe the dynamics of an action under some but not all conditions. We developed new formal models for learning partial action models and characterized the conditions under which they are successful.
- **Monte-Carlo Planning:** In many stochastic domains where compact action descriptions are not available, it is often possible to build a realistic simulation. Monte-Carlo simulations can be used to search many possible alternatives efficiently and find the best. We extended a Monte-Carlo Planning (MCP) approach called UCT to a number of complex domains that included durative actions and large action spaces and studied its effectiveness. We also developed a general computational infrastructure to study MCP.
- **Decision-Theoretic Assistance**: We developed a new formal decision-theoretic framework for computer assistants that help a human user by observing his overt behavior and take appropriate actions to optimize his expected utility. The key problem is to act rationally without the full knowledge of the goals of the human user. We studied the complexity of this problem under different conditions as well as provided an approximation bound for a reasonably effective heuristic solution.

Other broader impact and human resource contributions of our work included a week-long course on Monte-Carlo methods in AI for undergraduates and the graduation of two Ph.D. students.

2. Research Accomplishments

In this section, we describe our main research results in more detail**.**

2.1. Bayesian Reinforcement Learning

The problem of reinforcement learning is to learn how to act by taking actions in an environment and observing the outcomes and rewards. One of the main obstacles to reinforcement learning is that without strong prior knowledge the learning requires prohibitively large amount of experience before the performance is reasonable. More importantly, what is learned in one task does not generalize easily to the next task. Bayesian RL addresses both of these problems by biasing the learner with appropriate prior knowledge. The prior knowledge can be in the form a prior distribution over the dynamics of the domain (in model-based case) or on the policy, i.e., a parametric method of selecting actions given the state. The research problem we addressed is designing RL algorithms that allow transfer of policies from one task to another related task with only a reasonable amount of experience.

In particular, we developed a hierarchical Bayesian framework in which a prior distribution can be placed on policies, which can be improved over time by a form of Mante Carlo Markov Chain approach. The framework is hierarchical in the sense that the agents are divided into a variable number of classes according to what role they are good at playing. The algorithm automatically classifies agents into their classes, and assigns them roles, as well as creates new classes when needed. Importantly the classes and roles are shared between multiple tasks, so that the system is able to transfer the role-specific policies from one task to another. We applied this approach to a real-time strategy game domain and showed that our multiagent role discovery system outperforms previous best approaches. Thanks to the general Bayesian framework, the system can learn from expert demonstrations as well as its own random exploration. We showed that in hard tasks with a large number of agents, our system is able to transfer its policies learned from simpler tasks and perform quite well, while directly learning from the harder task fails completely [1].

One of the critical issues in Reinforcement Learning is the tradeoff between exploration and exploitation. Should an agent take the action that it knows is quite good, or should it explore a new action in the hope that it will uncover even better long-term rewards? We studied this directed exploration problem in the framework of Bayesian optimization. Bayesian Optimization (BO) works by carefully exploring the space of target policies by modeling the distribution of returns, i.e., expected long-term cumulative reward, of the policy as a function of the policy parameters. An effective heuristic in this paradigm is to probe the policy parameters which maximize the expected improvement in the returns. Previous work in applying Bayesian Optimization to RL treated the relationship between policy parameters and the returns as a black box, completely ignoring the role of the state-action trajectories traversed by the agent to obtain its cumulative reward. We made two contributions to enhance the power of BO to RL.

Our first contribution is a model-based approach, where we learn models of the domain dynamics from the trajectories. The models are used to simulate the actions and predict the rewards in Monte-Carlo fashion. However, the dynamics models may not be very accurate to predict the returns due to the complexity of the domain. Hence we separately model the errors in the Monte-Carlo returns with a Gaussian process. To take into account the approximate nature of the model and deal with cases when the model cannot be determined exactly, we used a weighted combination of the above model-based approach with zero mean model-free GP with an adjustable weight. We showed that this approach achieves significantly faster convergence than many model-based and model-free RL algorithms on a number of standard benchmark domains [2]. Our second contribution is a model-free approach where rather than using their parameters to compare the two policies, we employed a behavior-based kernel to compare them. The behavior-based kernel takes into account the trajectory information of the policies and is expected to better correlate with the returns. We showed that the behavior-based kernel gives comparable performance to the model-based approach. Both our methods outperform every other competitor there is including OLPOMDP, LSPI, DYNA-Q, and Q-learning [3].

2.2. Learning Task Hierarchies and Action Models

While reinforcement learning at the lowest level of primitive actions is inefficient, it can be made more efficient by predefining a hierarchy of tasks and subtasks and restricting the policies to this space. However, the current approaches that show the superiority of task hierarchies start from hand-designed hierarchies. Designing hierarchies by hand is not only time consuming, but is also unlikely to be successful in the long run because it requires non-trivial insight into the task. For many tasks that we like to automate, while there may well be experts who can perform the task or provide a few short demonstrations, they are unlikely to have the machine learning expertise or patience to design the task hierarchies. This leads to the problem of learning efficient task hierarchies which in turn make it possible to learn good policies quickly, starting from a small number of expert demonstrations of the task.

In previous work, we developed and an algorithm to learn task hierarchies from observed expert trajectories [4]. The algorithm works by hierarchically parsing the trajectory into contiguous chunks, where each chunk is as long as possible without increasing the set of features that is logically sufficient to guarantee its execution [5]. In more recent work, we improved this previous algorithm in a number of ways. First, the new algorithm is able to parse trajectories according to their causal structure rather than relying on the contiguity of actions in a subtask. This allowed us to learn from trajectories even when the trajectories do not follow a well-defined hierarchy. In particular, the trajectories may be generated by a random search or by a non-hierarchical planner. Second, we made our hierarchical architecture more general than previously studied MAXQ architecture by allowing tasks to be defined by sequences of subtasks, rather than by a single subtask followed by its parent task. Our HierGen system was able to learn from a richer class of trajectories where the previous approach failed, and provided a more complete solution to the problem of learning hierarchical task structures from observation [6].

Learning task hierarchies by analyzing the causal relations between actions required us to have correct models of the action. In the empirical work described above, we learned action models using adaptations of standard machine learning algorithms. However, this raised some fundamental questions about best ways to learn action models if one is only interested in planning using them, rather than requiring to predict the consequences of arbitrary actions in arbitrary states. Most of us will not be able to make good predictions when non-standard actions are taken, e.g., pressing the gas pedal and the break at the same time in a car or opening the engine, pulling out some wires and then driving. To formalize this intuition, we defined a notion of “adequate partial models” which are not sufficient to make sound predictions in all states, but are nevertheless sufficient for sound and complete planning for a goal distribution that one is interested in. We implemented and formalized means-ends-analysis-based planning with partial models [7].

We then introduced two new formal learning frameworks that captured the learning of partial models. The partial models are optimistic in the sense that they are at least as permissive as the true model. In the first Plan Prediction Mistakes framework, the learner is presented with problems to solve. The learner guarantees that adequate partial models are learned for the given goal distribution by making at most a polynomial number of mistakes. The partial models are optimistic, which guarantees that a sound planner using them makes only one kind of mistakes. In particular it might produce a plan that may not succeed, but never fails to return a plan when there is a correct plan. This allows the learner to learn from its own planning mistakes rather than requiring expert demonstrations ---- an important property to have for autonomous agents. We characterize sufficient conditions for exact learnability with at most polynomial number of mistakes in this framework. In the second Planned Exploration framework, which we also introduced, the learner designs its own planning problems to eliminate ambiguities in its models. Here we characterize the (stronger) conditions under which the learner succeeds to learn adequate models with only a polynomial number of planning attempts. In both cases, we present positive results for some concrete classes that generalize STRIPS operators with conditional effects and satisfy the sufficient conditions [8].

2.3. Monte-Carlo Planning

A big weakness of the most popular approaches to reinforcement learning is that they seek to completely eliminate search and reduce problem solving to reactive control. Unfortunately in challenging problems like military logistics and even relatively simple combinatorial puzzles and games like chess, this is demonstrably impossible. What is needed is the ability to improve solutions, given more time. Unfortunately the classical planning approaches try to find a sound plan or an optimal plan, but do not usually have control over the computation time. Moreover they require action models in a planning description language which are often unavailable. Monte-Carlo Planning (MCP) improves upon both planning and reinforcement learning by working with simulators rather than requiring action models, and by using more simulation time when available to improve the quality of the solutions. In particular, recently the UCT algorithm based on MCP was able to achieve impressive performance in the game of Go.

We found that an algorithm based on ensemble UCT performs very well on Klondike Solitaire winning more than 35% of games which is the current record [9]. Prompted by this success, we conducted a large empirical study of ensembles for MCP. In particular, we considered the potential benefits of using ensembles of trees produced by independent runs of the UCT algorithm for making decisions. We conducted experiments in 6 domains, encompassing a variety of characteristics, for a large range of ensemble and tree sizes. Our results demonstrated that given a multi-core architecture, the approach is extremely effective at improving performance per unit time. Given a single-core machine the approach is extremely effective at improving performance per unit space. Finally, we found that the approach did not typically lead to improved performance per unit time on a single core system, which weakly conflicts with prior results on Go and Solitaire. This work gave the first thorough evaluation of ensemble MCP, providing a much clearer idea about the general effectiveness of the approach [10].

2.4. Decision Theoretic Assistance

One of the goals of our work is to develop the theory needed to attack practical applications of planning technology in real world domains. Many such real world applications involve human-computer collaboration. This observation led us to more formally study the problem of human-computer interaction in a decision-theoretic setting. In particular, we formalized the problem of interactively assisting an agent whose goal is hidden, but whose actions are observable, as is typically the case in a variety of computer applications. We formalized this problem as Hidden Goal Markov Decision Process (HGMDP) and showed that although it is only a special case of Partially Observable Markov Decision Process (POMDP), it remains PSPACE-complete in the worst case. However, there exist some interesting special cases when the human agent is restricted to follow a stricter protocol. In particular, we introduced a model called Helper Action MDP (HAMDP), where the assistant's action must be accepted by the agent when it is helpful, and is otherwise ignored without cost. We showed that there is a simple myopic policy for HAMDPs which achieves a regret bounded by the entropy of the goal distribution when compared to an omniscient assistant. A variation of this policy was shown to achieve worst-case regret that is logarithmic in the number of goals for any goal distribution. We also derived a special case of HAMDPs which is NP-complete and another class where the complexity reduces to P [11, 12].

3. Infrastructure and Human Resource Developments

In addition to the above research accomplishments, we also contributed to the following infrastructure and human resource developments.

3.1. Development of Real-Time Strategy Game Infrastructure: We have significantly enhanced our infrastructure for an AI interface to the real-time strategy (RTS) game engine Stratagus. The new infrastructure, StratagusAI, is publically available at http://beaversource.oregonstate.edu/projects/stratagusai and implements a socket-based TCP/IP interface for playing the entire game via an AI agent. The interface provides access to all observations and commands that are available to a human player. The interface also provides mechanisms for restarting and running experiments in fast mode.

3.2. Development of a Simulation-Based Planning Library: We built a generic library for MCP algorithms. The library has standard APIs for simulators, environments, and planning algorithms. The library currently contains simulators for 8 diverse environments: Backgammon, Connect 4, Clue, Yatzhee, EWN, Biniax, Havannah, and Bird Conservation Management. It also includes a generic implementation of the full family of UCT algorithms that we have developed making it easy to experiment with different ensemble sizes and levels of sparseness. This infrastructure was used for the course described below and will be made broadly available in the near future.

3.3. Course on Monte-Carlo Methods in AI: We conducted our first spring break course on Monte Carlo methods in Artificial Intelligence in 2012 (MCAI-2012). We received 73 applications and accepted 19 students. We were able to recruit a diverse class of students including 9 women and 10 men; 2 native americans, 1 hispanic/latino, 1 native hawaiian, and 2 asians. We developed a class project around the problem of managing the endangered Red Cockaded Woodpecker. The course consisted of a mix of class room lectures by faculty including Fern and Tadepalli as well as hands-on lab assignments and a final project. Overall, the class was a big success, and we look forward to offering it again next spring. The students were fully supported by NSF under an infrastructure grant: http://web.engr.oregonstate.edu/mcai

3.4. Human Development: The current project led to the successful graduation of two Ph.D. students, Neville Mehta and Aaron Wilson.

4. Conclusions and Future Work

AI Planning and Reinforcement Learning attack the same broad problem of acting under uncertainty, but make very different assumptions and simplifications. In the current work, we explored the vast middle ground by considering variations and hybrids of both of these approaches, which allowed us to tackle a broader class of problems. We reformulated reinforcement learning in the hierarchical Bayesian framework which allowed us to generalize and transfer its results from one task to another and also devise more directed approaches to exploration. We employed causal analysis of trajectories to learn task hierarchies, which in turn are used to speed up reinforcement learning. We studied formal frameworks for learning action models from interaction, a problem which is mostly ignored in both planning and reinforcement literatures. We empirically studied variations of Monte-Carlo Planning, which facilitates anytime planning behavior using only a simulator, thus going beyond the classical approaches to reinforcement learning and planning. Finally our decision-theoretic assistance framework studied the theoretical properties of planning in the context of assistive systems, nicely illustrating the advantages of going beyond the dichotomy of planning and reinforcement learning.

One important problem we have not tackled in this project is that of *speedup learning*, or improving the anytime performance of planners with experience. With the availability of reasonably good anytime planning algorithms such as UCT, it is vitally important that the learning algorithms improve their planning performance over time by learning appropriate heuristics or generalized value functions. This is a woefully understudied problem especially in the context of stochastic and adversarial environments. We believe that many of the algorithmic ideas studied in this project including the learning of action models and analyzing their interaction, hierarchical Bayesian models and Bayesian inference, and variations of Monte-Carlo Planning would be useful in this endeavor.

References

[1] Wilson, A., Fern, A., and Tadepalli, P., Bayesian Policy Search for Multi-Agent Role Discovery. In *National Conference on Artificial Intelligence*, 2010.

[2] Wilson, A., Fern, A., and Tadepalli, P., Incorporating Domain Models into Bayesian Optimization for Reinforcement Learning, In *European Conference on Machine Learning*, 2010.

[3] Wilson, A., Fern, A. and Tadepalli, P., A Behavior Based Kernel for Policy Search via Bayesian Optimization in *Workshop on Planning and Acting with Uncertain Models* at ICML, 2011.

[4] Mehta, N., Ray, S., Tadepalli, P., and Dietterich, T., Automatic Discovery and Transfer of MAXQ Hierarchies, In *International Conference on Machine Learning*, 648--655, 2008.

[5] Mehta, N., Ray, S., Tadepalli, P., and Dietterich, T. G., Automatic Discovery and Transfer of Task Hierarchies in Reinforcement Learning, *AI Magazine*, 32(1): 35-50, 2011.

[6] Mehta, N., Hierarchical Structure Discovery and Transfer in Sequential Decision Problems, Ph.D. Dissertation, Oregon State University, 2011.

[7] Mehta, N., Tadepalli, P., and Fern, A. Learning and Planning with Partial Models**,** Workshop on Learning Structured Knowledge from Observations, at *IJCAI*, 2009.

[8] Mehta, N., Tadepalli, P., Fern, A.: Autonomous Learning of Action Models for Planning, In *Advances in Neural Information Processing Systems*, 2465-2473, 2011.

[9] Bjarnason, R., Fern, A. and Tadepalli, P., Lower Bounding Klondike Solitaire with Monte-Carlo Planning. in *International Conference on Automated Planning and Scheduling*, 2009.

[10] Lewis, P. and Fern, A. Ensemble Monte-Carlo Planning: An Empirical Study, In *International Conference on Automated Planning and Scheduling*, 2011.

[11] Fern, A. and Tadepalli, P. A Computational Decision Theory for Interactive Assistants, In the workshop on *Interactive Decision Theory and Game Theory* at AAAI, 2010.

[12] Fern, A. and Tadepalli, P. A Computational Decision Theory for Interactive Assistants. In *Advances in Neural Information Processing System*, 2010.

Acknowledgments

We gratefully acknowledge the Army Research Office, the Army Research Laboratory, and our program manager, Purush Iyer, for supporting this work under grant number W911NF-09-1-0153.